2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-5300-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/CVPR52733.2024.00699

Neural Implicit Morphing of Face Images

Guilherme Schardong[1,*] Tiago Novello[2,*] Hallison Paz[2] Iurii Medvedev[1] Vinícius da Silva[3]
Luiz Velho[2] Nuno Gonçalves[1,4]

[1] Institute of Systems and Robotics, University of Coimbra
[2] Institute of Pure and Applied Mathematics
[3] Tecgraf, Pontifical Catholic University of Rio de Janeiro
[4] Portuguese Mint and Official Printing Office

Abstract

Face morphing *is a problem in computer graphics with numerous artistic and forensic applications. It is challenging due to variations in pose, lighting, gender, and ethnicity. This task consists of a* ***warping*** *for feature alignment and a* ***blending*** *for a seamless transition between the warped images. We propose to leverage* ***coord-based neural networks*** *to represent such warpings and blendings of face images. During training, we exploit the smoothness and flexibility of such networks by combining energy functionals employed in classical approaches without discretizations. Additionally, our method is* ***time-dependent****, allowing a continuous warping/blending of the images. During morphing inference, we need both direct and inverse transformations of the time-dependent warping. The first (second) is responsible for warping the target (source) image into the source (target) image. Our neural warping stores those maps in a single network dismissing the need for inverting them. The results of our experiments indicate that our method is competitive with both classical and generative models under the lens of image quality and face-morphing detectors. Aesthetically, the resulting images present a seamless blending of diverse faces not yet usual in the literature.*

1. Introduction

Image warping is a continuous transformation mapping points of the image support to points in a second domain. The process of warping an image has applications ranging from correcting image distortions caused by lens or sensor imperfections [9] to creating distortions for artistic/scientific purposes [5]. Warping finds a special application in creating *image morphings* [10], where it is used to align corresponding features. By gradually aligning the image features using the warping, we obtain a smooth transition between them.

We assume the warpings to be parameterized by smooth maps. Besides obtaining smooth transitions, this allows us to use its derivatives to constrain the deformation, such as approximating it as a minimum of a *variational problem*. Feature alignment can be specified using *landmarks* to establish correlations between two images.

In this work, we use *coord-based neural networks*, which we call *neural warpings*, to parameterize image warpings. This approach enables us to calculate the derivatives in closed form, eliminating the need for discretization. We also employ a time parameter, to represent smooth transitions. By incorporating the derivatives into the loss function, we can regularize the network and easily add constraints by summing additional terms. To train a neural warping, we propose a *loss function* consisting of two main terms. First, a *data constraint* ensures that the warping fits the given keypoint correspondences. Second, we *regularize* the neural warping using the *thin-plate* energy to minimize distortions.

We use neural warping to model *time-dependent* morphings of face images, thus aligning the image features over time. Afterward, we explore the flexibility of coord-based neural networks to define three blending techniques. First, we blend the aligned image warpings in the *signal domain* using point-wise interpolation. Second, we propose to blend the image warpings in the *gradient-domain* of the signals. For this, we introduce another neural network to represent the morphing and train it to satisfy the corresponding variational problem. If the target faces have different semantics, we cannot adequately blend the warped images in the signal/gradient domain; therefore, we propose a third option: blending using generative methods. In other words, we propose to use a *generative mixing*: we embed the image warpings in a *latent space* of some generative model, then we interpolate the resulting embedding and project it back to the image space. We present experiments using Diffusion Auto-encoders (diffAE) [27].

Our contributions can be summarized as follows:

[*] These authors contributed equally to this work.
Project page: https://schardong.github.io/ifmorph

2575-7075/24/$31.00 ©2024 IEEE
DOI 10.1109/CVPR52733.2024.00699

- The introduction of a time-dependent **neural warping** which encodes in a single network the *direct* and *inverse* transformations needed to align two images along time. We use the warping to transport the images and their derivatives from the initial states to intermediate times.
- The neural network is **smooth**, both in space and time, enabling the use of its derivatives in the loss function. We exploit it to define an implicit regularization using the *thin-plate* energy which penalizes distortions. Thus, the landmarks follow a path that minimizes this energy instead of a straight line, as in classical approaches.
- The neural warping model is **compact**. We achieved accurate warping using a MLP composed of a single hidden layer with 128 neurons, although our ablation studies indicate that smaller networks would work for specific cases.
- We blend the resulting aligned image warpings to define a time-dependent **morphing**, distinguishing it from current methods that focus on a single blend. For the case of blending in the gradient-domain, we use another neural network (**neural morphing**). For the **generative morphing**, we embed the warpings in a latent space, interpolate the resulting curves, and project it back to image space.

2. Related Works

The first algorithms for face morphing were simple *cross-dissolves*, i.e., pixel interpolation between target images [34]. However, the resulting morphings are substandard unless the images are aligned, resulting in artifacts. To overcome this, *mesh-based* alignment was used before the interpolation stage, shifting the complexity to the image alignment. Beier and Neely [2] further refined the process using line correspondences and an interface to align them. Liao et al. [18] exploited halfway domains, *thin-plate* splines, and *structural similarity* to create a discrete vector field to warp the images.

The above morphing approaches are landmark-based, as is ours. Recently, generative methods, such as StyleGANs [13–15] and diffAE [27], have also been used to interpolate between faces. In contrast to these methods, ours is *smooth* in both time and space, as we have a differentiable curve tracking the path of each image point during warping. Moreover, our approach exploits the recent *implicit neural representations*, which employ coord-based neural networks [32] to parameterize the images. Hence, we eliminate the need for interpolation and image resampling. This approach has also been used in the context of generative models [1] and multiresolution image representation [25].

Furthermore, by implicitly representing the images, we obtain their *derivatives in closed form* through automatic differentiation, which is not possible with previous landmark and generative approaches. This allows efficient use of the gradient during the training/analysis. Moreover, composing the warping and images results in the warped images with gradients given by the product of the warping Jacobian and the image gradient.

An important step in our warping is the incorporation of the time variable as input of the neural warping. Combined with the above advantages, this enables the creation of continuous, smooth, and compact warpings. This also allows us to constrain the landmark paths over time by minimizing distortions, unlike classical methods.

Regarding StyleGANs and diffusion models, StyleGANs create a latent space of images. Thus, the blending between two faces is an interpolation of the corresponding projected codes in the latent space. It produces high-quality images, although their embedding is not necessarily invertible. Therefore there is no guarantee that the blendings will be strictly of the desired faces [27].

On the other hand, diffAE uses a learnable encoder to discover the high-level semantics of the image and *denoising the implicit diffusion model* [33] to decode and model stochastic variations. Unlike StyleGANs that depend on error-prone inversion, diffAE encodes the image without an additional optimization step. The outputs of the target images are close to the originals, which is desirable for blending.

Additionally, StyleGANs may not satisfy the property of blending the target faces over time since features of other faces (from the training dataset) can appear in the intermediate frames (Fig 6). We note no such problem using the generative blending of diffAE. That is why we use it as an example of neural blending in our framework.

Note that generative models do not align image features over time, as they do not model any warping of the image domains. Instead, they perform a *generative blending* between the images. Furthermore, such models rely on latent code interpolations, and while they can blend the target images, they lack temporal coherence (see the video in the supp. mat.). Also, these models consider the face images to be aligned by placing the eyes and mouths at fixed locations in the image support. Thus restricting face interpolation to a specific case, where eyes and mouths are fixed over time.

Our morphing approach does not suffer from the said issues, since it disentangles the warping from the blending, thus allowing for different blendings, such as Poisson image blending and generative blending. For instance, the output of our neural warping can serve as input for a generative blending, enabling faces in different positions, ensuring temporal coherence, and tracking the path of each point in the image support over time (see Fig 7 and the video in supp. mat.).

Morphing enables the creation of synthetic faces remarkably similar to real ones, known as "face-morphing attack". These techniques have captured the attention of the biometrics community, resulting in a body of works dedicated to detecting such attacks [8, 28]. Our method has the potential to generate new datasets, enhancing the effectiveness of these detection systems. In biometrics, the production and identification of morphed images are primarily concerned with images that comply with the International Civil Aviation Organization (ICAO) standards [4, 11]. Morphing can create images that merge the biometric identifiers of multiple individuals, resulting in a facial image that could match several people. Such images in official identification documents pose a significant threat, as they undermine the fundamental principle of biometric verification: one document should correspond to an unique identity.

3. Methodology

3.1. Background and Notation

We represent an *image* by a function $\mathrm{I} : \Omega \subset \mathbb{R}^2 \to \mathcal{C}$, where Ω is the image *support* and $\mathcal{C}$ is the *color space*, and parameterize it using a (coord-based) neural network $\mathrm{I}_\theta : \mathbb{R}^2 \to \mathcal{C}$ with parameters θ. To train the *neural image* I_θ such that it approximates I, we can optimize $\int_\Omega (\mathrm{I} - \mathrm{I}_\theta)^2 \, dx$. This work explores *coord-based neural networks* to morph *neural images* using a novel neural *warping* approach.

We assume that a coord-based neural network is a *sinusoidal* multilayer perceptron (MLP) [17, 24, 32] $f_\theta(p) : \mathbb{R}^n \to \mathbb{R}^m$ defined as the composition $f_\theta(x) = W_d \circ f_{d-1} \circ \cdots \circ f_0(x) + b_d$ of d *sinusoidal layers* $f_i(x_i) = \sin(W_i x_i + b_i) = x_{i+1}$, where $W_i \in \mathbb{R}^{n_{i+1} \times n_i}$ are the weight matrices, and $b_i \in \mathbb{R}^{n_{i+1}}$ are the biases. The union of these parameters defines θ. The integer d is the *depth* of f_θ and n_i are the layers *widths*.

The MLP f_θ is smooth because its layers are composed of smooth maps, and we can compute its derivatives in closed form using automatic differentiation. This property plays an important role in our method since it allows using derivatives for implicit regularization of the warpings and morphings.

3.2. Neural Morphing

This section introduces the *neural morphing* of two images. It consists of a *neural warping* to align the features of the image and a *neural blending* of the resulting warped images.

Specifically, let $\mathrm{I}_0, \mathrm{I}_1 : \mathbb{R}^2 \to \mathcal{C}$ be two neural images, we represent their *neural morphing* using a (time-dependent) neural network $\mathcal{I} : \mathbb{R}^2 \times [0, 1] \to \mathcal{C}$ subject to $\mathcal{I}(\cdot, i) = \mathrm{I}_i(\cdot)$, for $i = 0, 1$. Thus, for each t we have an image $\mathcal{I}(\cdot, t)$, and varying t results in a video interpolating I_i. To define the morphing $\mathcal{I}$, we **disentangle** the spatial deformation (*warping*), used to align the corresponding *features* of I_i along the time, from the *blending* of the resulting warped images.

For the warping, we use pairs of *landmarks* $\{p_j, q_j\}$, with j being the *landmark index*, sampled from the domains of I_0 and I_1 providing feature correspondences. Then, we seek a warping $\mathbf{T}: \mathbb{R}^2 \times [-1, 1] \to \mathbb{R}^2$ satisfying the *data constraints*:

- The curves $\mathbf{T}(p_j, t)$ and $\mathbf{T}(q_j, t-1)$, with $t \in [0, 1]$, has p_j and q_j as end points;
- For each $t \in (0, 1)$, we require $\mathbf{T}(p_j, t) = \mathbf{T}(q_j, t-1)$.

Thus, the values $\mathrm{I}_0(p_j)$ and $\mathrm{I}_1(q_j)$ can be blended along the path $\mathbf{T}(p_j, t)$. In points $x \neq p_j$, we employ the well-known *thin-plate* energy to force the transformations to be as affine as possible. The resulting network $\mathbf{T}$ deforms I_i along the time resulting in the *warpings* $\mathcal{I}_i : \mathbb{R}^2 \times [0, 1] \to \mathcal{C}$ defined as:

$$\mathcal{I}_i(x, t) := \mathrm{I}_i\big(\mathbf{T}(x, i - t)\big). \qquad (1)$$

Fig 1 illustrates the warpings $\mathcal{I}_i$. Given a point (x, t), to evaluate x in image I_i we move it to time $t = i$, for $i = 0, 1$, which is done by $x_i := \mathbf{T}(x, i - t)$. Note that for x_0 and x_1, we need the inverse and direct transformations of $\mathbf{T}$ (in red/blue) since it employs negative and positive time values.

Then we obtain the image values by evaluating $\mathrm{I}_i(x_i)$. Moreover, we can move a vector v_i at x_i to x, at time t, considering the product $v_i \cdot \mathrm{Jac}(\mathbf{T}(x, i - t))$, where Jac is the Jacobian. In Section 3.4, we use such property and consider $v_i = \nabla \mathrm{I}_i(x_i)$ to blend the images in the *gradient domain*.

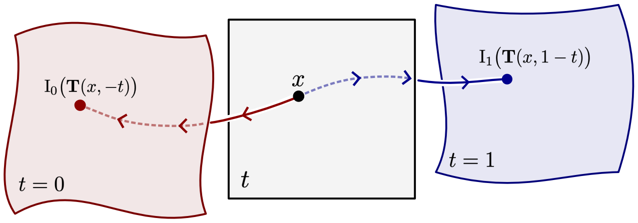

$\mathrm{I}_0(\mathbf{T}(x, -t))$
x
$\mathrm{I}_1(\mathbf{T}(x, 1-t))$
$t = 0$
t
$t = 1$

Figure 1. Schematic illustration of the neural warping $\mathbf{T}$ being used to aligning the initial images I_i

We blend the resulting aligned warpings $\mathcal{I}_i$ to define the desired morphing $\mathcal{I} : \mathbb{R}^2 \times [0, 1] \to \mathcal{C}$. We consider three blending approaches: a simple linear interpolation $\mathcal{I} = (1-t)\mathcal{I}_0 + t\mathcal{I}_1$, blending in the *gradient domain* using the Poisson equation, and *generative blending* using diffAE. Section 3.4 presents these approaches in detail.

The following steps summarize the procedure of morphing two images I_i:

- Extract **key points** $\{p_j, q_j\}$ in the domains of the face images I_0 and I_1, providing feature correspondence.
- Define and train the **neural warping T** : $\mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$ to align the key points $\{p_j, q_j\}$ while penalizing distortions using the thin-plate energy. This produces the image warpings $\mathcal{I}_i$ that align the features of I_i along time;
- Blend $\mathcal{I}_i$ to define the **morphing** $\mathcal{I} : \mathbb{R}^2 \times \mathbb{R} \to \mathcal{C}$ of I_i. We consider two representations for $\mathcal{I}$. First, we use a sinusoidal MLP and exploit its flexibility to train in the *gradient domain*. Second, we embed $\mathcal{I}_i$ in the latent space of diffAE resulting in two curves, then $\mathcal{I}$ is given by interpolating these curves and projecting back to image space.

3.3. Neural warping

This section presents the *neural warping*, a neural network that aligns features of the target images along time. Precisely, we model it using a sinusoidal MLP $\mathbf{T} : \mathbb{R}^2 \times [-1, 1] \to \mathbb{R}^2$, and require the following properties:

- $\mathbf{T}(\cdot, 0)$ is the *identity* (Id);
- For each $t \in [-1, 1]$, we have that $\mathbf{T}_{-t}$ is the *inverse* of $\mathbf{T}_t$.

The corresponding deformation of an image $\mathrm{I}:\mathbb{R}^2\to\mathcal{C}$ by $\mathbf{T}$ is defined using $\mathscr{I}(\cdot,t)=\mathrm{I}\circ\mathbf{T}(\cdot,-t)$ which uses the inverse $\mathbf{T}_{-t}$ of $\mathbf{T}_t$. That is one of the reasons we require the inverse property. In fact, if $\mathbf{T}$ holds such a property, there is no need to invert the *direct* warp $\mathbf{T}_t$, which is a difficult task in general. For simplicity, we say that $\mathscr{I}$ is a *warping* of I. Note that at $t=0$, we have $\mathscr{I}(\cdot,0)=\mathrm{I}$ because $\mathbf{T}(\cdot,0)=\mathrm{Id}$. Thus, $\mathscr{I}$ evolves the initial image I along time.

We could avoid using the inverse map $\mathbf{T}_{-t}$ by employing a sampling $\{\mathrm{I}_{ij}\}$ of I on a regular grid $\{x_{ij}\}$ of the image support. Then, $\{\mathrm{I}_{ij}\}$ are samples of the warped image $\mathrm{I}\circ\mathbf{T}_{-t}$ at points $\{\mathbf{T}_t(p_{ij})\}$. However, this approach has the drawbacks of resampling $\mathrm{I}\circ\mathbf{T}_{-t}$ in a new regular grid which can result in *holes* and relies on interpolation techniques. Our method avoids such problems since it will be trained to fit the property $\mathbf{T}_t\circ\mathbf{T}_{-t}=\mathrm{Id}$ for $t\in[-1,1]$.

Observe that, for each t, the map $\mathbf{T}_t$ approximates a *diffeomorphism* since it is a smooth sinusoidal MLP with an inverse also given by a sinusoidal MLP $\mathbf{T}_{-t}$ since $\mathbf{T}_t\circ\mathbf{T}_{-t}=\mathrm{Id}$.

3.3.1 Loss function

Let $\mathrm{I}_0,\mathrm{I}_1:\mathbb{R}^2\to\mathcal{C}$ be neural images and $\{p_j,q_j\}$ be the *source* and *target* points sampled from the supports of I_0 and I_1 that provide feature correspondences. Let $\mathbf{T}:\mathbb{R}^2\times\mathbb{R}\to\mathbb{R}^2$ be a sinusoidal MLP, we train its parameters θ so that $\mathbf{T}$ approximates a warping aligning the key points p_j and q_j along time. For this, we use the following loss functional.

$$\mathscr{L}(\theta)=\mathscr{W}(\theta)+\mathscr{D}(\theta)+\mathscr{T}(\theta). \qquad (2)$$

Where $\mathscr{W}(\theta)$, $\mathscr{D}(\theta)$, $\mathscr{T}(\theta)$ are the *warping*, *data*, and *thin-plate* constraints. $\mathscr{W}(\theta)$ requires the network $\mathbf{T}$ to satisfy the identity and inverse properties of the warping definition.

$$\mathscr{W}(\theta)=\underbrace{\int_{\mathbb{R}^2}\|\mathbf{T}(x,0)-x\|^2dx}_{\text{Identity constraint}}+\underbrace{\int_{\mathbb{R}^2\times\mathbb{R}}\|\mathbf{T}(\mathbf{T}(x,t),-t)-x\|^2dxdt}_{\text{Inverse constraint}}. \qquad (3)$$

The *identity* constraint forces $\mathbf{T}_0=\mathrm{Id}$ and, the *inverse* constraint asks for $\mathbf{T}_{-t}$ to be the inverse of $\mathbf{T}_t$ for all $t\in\mathbb{R}$.

The *data constraint* $\mathscr{D}(\theta)$ is responsible for forcing $\mathbf{T}$ to move the source points p_j to the target points q_j such that their paths match along time. For this, we simply consider:

$$\mathscr{D}(\theta)=\int_{[0,1]}\|\mathbf{T}(p_j,t)-\mathbf{T}(q_j,1-t)\|^2\,dt \qquad (4)$$

Note that $\mathscr{D}$ is asking for $\mathbf{T}(p_j,1)=q_j$ and $\mathbf{T}(q_j,-1)=p_j$ because at the same time $\mathscr{W}$ is forcing the identity property. Moreover, it forces $\mathbf{T}(p_j,t)=\mathbf{T}(q_j,1-t)$ along time, thus, as observed at the beginning of this section, this is the required property for the key points $\{p_j,q_j\}$ be aligned along time. Since we assume $\mathbf{T}$ to be a sinusoidal MLP, the resulting warping provides a smooth deformation that moves the source points to the target points.

However, $\mathscr{D}$ does not add restrictions on points other than the source and target points. Even assuming $\mathbf{T}$ to be smooth the resulting warping would need some regularization, such as minimizing distortions. For this, we propose a *regularization* which penalizes distortions of the transformations $\mathbf{T}_t$ using the well-known the *thin-plate* energy [3, 9]:

$$\mathscr{T}(\theta)=\int_{\mathbb{R}^2\times\mathbb{R}}\|\mathbf{Hess}(\mathbf{T})(x,t)\|_F^2\,dxdt. \qquad (5)$$

$\mathscr{T}$ regularizes $\mathbf{T}$ and works as a bending energy term penalizing deformation, at each space-time point (x,t), based on the derivatives of $\mathbf{T}$. This helps eliminate global effects that may arise from considering only data and warping constraints. It is important to note that we have incorporated the time variable into the thin-plate energy $\mathscr{T}$.

By using a sinusoidal MLP to model $\mathbf{T}$ and training it with $\mathscr{W}$ while regularizing with the thin-plate energy, we achieve robust warpings, see Fig 2 for an alignment between two images, for more detail see the experiments in Sec 4.

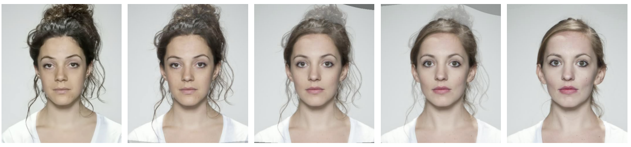


Figure 2. A neural warping $\mathbf{T}$ continuously aligning two face images along time. We use $\mathbf{T}$ to create their aligned warpings $\mathscr{I}_i$. The morphing $(1-t)\mathscr{I}_0+t\mathscr{I}_1$ was sampled at $t=0, 0.25, 0.5, 0.75, 1$.

Additionally, we perform experiments to assess the impact of each term $\mathscr{W}$, $\mathscr{D}$, $\mathscr{T}$ to understand their importance during the training of $\mathbf{T}$. We found out that the thin-plate constraint $\mathscr{T}$ is crucial. Also, as expected without the data constraint $\mathscr{D}$ we can not align the image features. The warping constraint has less influence, acting mostly on finer details. That was an interesting finding implying that the warping properties are being enforced by $\mathscr{T}$. This is probably due to the fact that $\mathscr{D}$ forces such property along the feature paths and $\mathscr{T}$ asks for the deformation to be minimized in $\mathbb{R}^2\times[-1,1]$. Fig 3 illustrates the experiment.

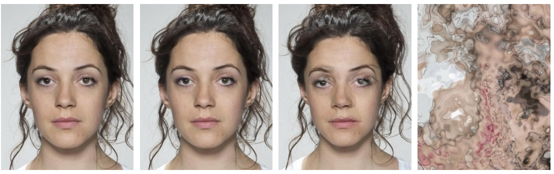


Figure 3. Loss term impact experiment. From the left: results without the inverse, identity, data, and thin-plate constraints.

3.4. Neural Blending

Let $\mathrm{I}_i:\mathbb{R}^2\to\mathcal{C}$ be two neural images and $\mathbf{T}:\mathbb{R}^2\times\mathbb{R}\to\mathbb{R}^2$ be a neural warping aligning their features. Specifically, the images I_i are deformed by $\mathbf{T}$ along time and Eq 1 gives the corresponding warpings $\mathcal{I}_i(x,t) = \mathrm{I}_i\big(\mathbf{T}(x, i-t)\big)$. Then, we blend $\mathcal{I}_i$ or their derivatives to construct a morphing $\mathcal{I}: \mathbb{R}^2 \times \mathbb{R} \to \mathcal{C}$ of the initial images I_i. A naive blending approach could be defined directly from $\mathcal{I}_i$ by interpolating using $\mathcal{I}(x,t) = (1-t)\mathcal{I}_0(x,t) + t\mathcal{I}_1(x,t)$. Thus, at $t=0$ and $t=1$, we obtain $\mathcal{I}_0$ and $\mathcal{I}_1$, respectively (See Fig 2). Note that $\mathcal{I}$ is a smooth function both in time and space.

3.4.1 Blending in the gradient domain

Interpolating I_i does not allow us to keep parts of one of the images unchanged during the morphing, e.g. the complement region of the face. To address these issues, inspired by the *Poisson image editing* technique [26], we propose to blend I_i by solving a *boundary value problem* in $\mathbb{R}^2 \times \mathbb{R}$ to handle smooth animations and model $\mathcal{I}$ by a neural network.

We use the Jacobians $\text{Jac}(\mathcal{I}_i)$ of the warpings $\mathcal{I}_i$ to train $\mathcal{I}$. We restrict the morphing support to $S{=}[-1,1]^2{\times}[0,1]$, with $[-1,1]^2$ representing the image domain and $[0,1]$ is the time interval. Let $\Omega \subset S$ be an open set used for blending $\mathcal{I}_i$, such as the interior of the face path, and let $\mathcal{I}^*\colon S \to \mathbb{R}$ be a known function on $S - \Omega$ (it could be either $\mathcal{I}_0$ or $\mathcal{I}_1$). Finally, let U be a matrix field obtained by blending $\text{Jac}(\mathcal{I}_i)$, for example, $U = (1-t)\text{Jac}(\mathcal{I}_0) + t\text{Jac}(\mathcal{I}_1)$. A common way to extend $\mathcal{I}^*$ to Ω is by solving:

$$\min \int_{\Omega} \|\text{Jac}(\mathcal{I}) - U\|^2 dxdt \text{ subject to } \mathcal{I}|_{S-\Omega} = \mathcal{I}^*|_{S-\Omega}. \quad (6)$$

We propose to use this variational problem to define the following loss function to train the parameters θ of $\mathcal{I}$.

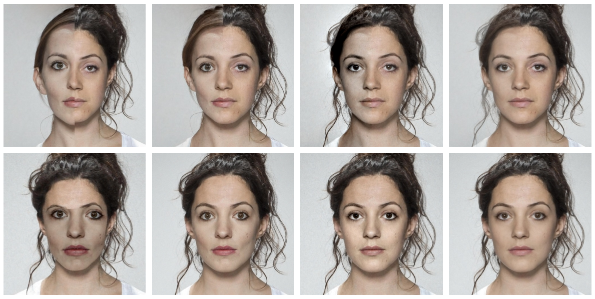
$$\mathcal{M}(\theta) = \underbrace{\int_{\Omega} \|\text{Jac}(\mathcal{I}) - U\|^2 \, dxdt}_{\mathcal{C}(\theta)} + \underbrace{\int_{S-\Omega} (\mathcal{I} - \mathcal{I}^*)^2 dxdt}_{\mathcal{B}(\theta)}. \quad (7)$$

The *cloning term* $\mathcal{C}(\theta)$ fits $\mathcal{I}$ to the primitive of U in Ω, and the *boundary constraint* $\mathcal{B}(\theta)$ forces $\mathcal{I} = \mathcal{I}^*$ in $S{-}\Omega$. Thus, $\mathcal{M}$ trains $\mathcal{I}$ to *seamless clone* the primitive of U to $\mathcal{I}^*$ in Ω. Unlike classical approaches that rely on pixel manipulation, seamless cloning operates on the image gradients.

Since the images I_i contain faces and $\mathbf{T}$ aligns their features, we define Ω as the path of the facial region over time. Specifically, let Ω_0 be the region containing the face in I_0, define Ω by warping Ω_0 along time using $\mathbf{T}$, i.e., $\Omega = \cup_{t\in[0,1]} \mathbf{T}_t(\Omega_0)$. Note that the deformation of Ω_0 uses the direct deformation $\mathbf{T}_t$ while the warped image $\mathcal{I}_0$ uses the inverse $\mathbf{T}_{-t}$. The use of both inverse/direct deformations encoded in our neural warping avoids the need to compute inverses at inference time. Finally, for each t, $\mathbf{T}$ aligns the faces I_i in the region $\mathbf{T}_t(\Omega_0)$. Thus, $\mathcal{M}$ trains $\mathcal{I}$ to morph the face in I_0 into the face in I_1 while cloning the result to $\mathcal{I}_0$ on $S - \Omega$.

Besides choosing U as a linear interpolation of $\text{Jac}(\mathcal{I}_i)$, which we call the *averaged seamless cloning* case, we could choose $U = \text{Jac}(\mathcal{I}_1)$ and $\mathcal{I}^* = \mathcal{I}_0$. So, the resulting loss function $\mathcal{M}$ forces $\mathcal{I}$ to *seamless clone* the face $\mathcal{I}_1$ to the corresponding region of $\mathcal{I}_0$.

It may be desirable to combine features of $\mathcal{I}_i$, however an interpolation of $\text{Jac}(\mathcal{I}_i)$ can lead to loss of details. To avoid it, we extend the approach in [26], which allows mixing the features of both images. At each (x,t), we retain the stronger of the variations in the warpings by choosing $U{=}\text{Jac}(\mathcal{I}_0)$ if $\|\text{Jac}(\mathcal{I}_0)\| > \|\text{Jac}(\mathcal{I}_1)\|$, and $U{=}\text{Jac}(\mathcal{I}_1)$, otherwise. The resulting loss function $\mathcal{M}$ forces $\mathcal{I}$ to learn a *mixed seamless clone* of $\mathcal{I}_i$. Fig 4 shows examples of neural blending.



No warping seamless cloning average cloning mixed cloning

Figure 4. Comparing different neural blendings of two faces I_i. Line 1/2 shows examples of cloning the half-space region of I_1 into I_0. In Column 1 we do not align the image landmarks, the remaining columns use our neural warping for the alignment. Column 2 uses $U{=}\text{Jac}(\mathcal{I}_1)$ and $\mathcal{I}^*{=}\mathcal{I}_0$ in the neural blending. Columns 3 and 4 applies the mixed and normal seamless clone respectively.

3.4.2 Blending using generative models

Generative models may be used to interpolate faces. However, they do not ensure feature alignment, only provide a blending of the images. To overcome this issue, we use our neural warping to align the face features and a generative blending to combine the resulting warped images over time. Sec. 4.2 presents experiments with this approach.

Specifically, let I_i be neural images representing two faces and $\mathbf{T}$ be a neural warping aligning their features. Again, the images I_i are deformed by $\mathbf{T}$ along time resulting in the image warpings $\mathcal{I}_i$. Recall that, for each $t \in [0,1]$, we have that the faces in $\mathcal{I}_0(t)$ and $\mathcal{I}_1(t)$ have their features aligned. Let $\mathcal{E}$ and $\mathcal{D}$ be the *encoder* and *decoder* of a generative model. We embed $\mathcal{I}_i$ in the latent space which results in the *code curves* $c_i(t) = \mathcal{E}\big(\mathcal{I}_i(\cdot,t)\big)$. Then, we interpolate the curves directly in the latent space and the desired generative morphing is given by projecting the resulting curve to the image space using the decoder $\mathcal{D}$:

$$\mathcal{I}(\cdot,t) := \mathcal{D}\Big((1-t)c_0(t) + tc_1(t)\Big). \quad (8)$$

With the *generative morphing* $\mathcal{I}$ we have the feature correspondence along time and their path explicitly. We use it to improve the temporal coherence in generative approaches.

In practice, we employ diffAE [27] since, unlike GANs that depend on error-prone inversion, it encodes the input and produces high-quality output without an optimization step. Moreover, the output of the target images is close to the originals, i.e. $\mathcal{I}(i) \approx \mathrm{I}_i$, which is desirable for the morphing task. Also, note that to blend images using diffAE we have to interpolate between two-part codes with a semantic and a stochastic part.

Fig 5 shows a comparison between the generative morphing and a pure diffAE applied to I_i. Line 1 presents samples of the generative morphing $\mathcal{J}(\cdot, t)$. In Line 2, we simply interpolate between the codes of I_i. Note that the generative morphing offers smoother transitions between corresponding features; see the video in the supplementary material.

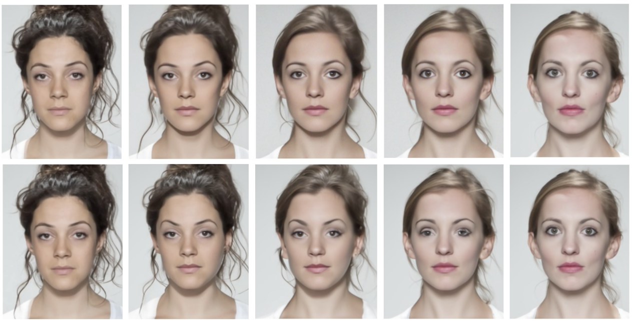


Figure 5. Generative morphing. Line 1 presents a morphing between two faces using the generative morphing (neural warping + diffAE). Line 2 shows the results of diffAE using no alignment.

This experiment does not employ the pre-processing step of fixing the eyes and mouth in the image support. This step is common in generative approaches and relies on DLib [16, 29] to detect facial features. For the experiment using this alignment, refer to Fig 6. However, such dependence on generative models forces the eyes and mouths to remain fixed in the image support over time. Hence, we cannot morph between roto-translated images.

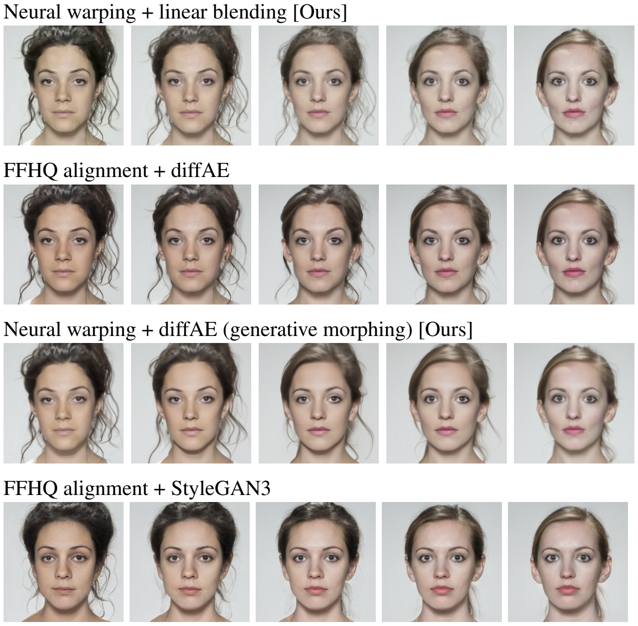
4. Experiments and Discussions

In the experiments, we used small sinusoidal MLPs consisting of a single hidden layer with 128 neurons to parametrize the neural warpings. However, our ablation study indicated that smaller networks also works, see the supp. material. This shows that our representation is compact and robust for time-dependent warpings. The network initialization follows the definitions in [32]. Additionally we use DLib [16, 29] for landmark detection. For the experiments, StyleGAN3 was fine-tuned with images from the FRLL dataset for 312 epochs, while diffAE was used directly from the authors' repository (model FFHQ256, autoencoding only).

4.1. Qualitative comparisons

We assess our approach regarding the visual quality of both warping/blending of faces. Fig 6 shows our neural warping with linear blending, diffAE with FFHQ alignment, neural warping and diffAE, and StyleGAN3 with FFHQ alignment. Note that unlike StyleGAN3, diffAE provides a close, although blurred, reconstruction of the target.

In Fig 6, diffAE (Line 2) produces a shadow in the forehead/hair transition area for images with $t = 0.5, 0.75$. It also creates a hole in the subject's left earlobe. These issues are missing when using our neural warping for alignment (Line 3). Another point of note is the face similarity between neural warping + diffAE (Line 3) and neural warping + linear blending (Line 1). This is due to the temporal coherence added by time-dependent alignment given by the warping. Thus, the generative morphing produces intermediate faces closer to the targets when compared to employing FFHQ alignment. Moreover, since StyleGAN3 does not reproduce the target faces from the latent code projections, the blendings are generating faces unrelated to the originals.



Neural warping + linear blending [Ours]

FFHQ alignment + diffAE

Neural warping + diffAE (generative morphing) [Ours]

FFHQ alignment + StyleGAN3

Figure 6. Morphing comparisons of our method and generative approaches (neural warping + linear blending, diffAE, neural warping + diffAE, and StyleGAN3). Columns 1 and 5 are the target faces, while the three middle columns are blendings for $t = 0.25, 0.5, 0.75$. The original images are the ends of Line 1.

As shown in Fig 5, FFHQ alignment is necessary for interpolating faces; otherwise, it produces visual artifacts. This is because generative models do not perform warping of facial features; instead, they blend them. Thus, we cannot use such methods for morph faces in different poses. However, we observe that we can use neural warping for this task. Fig 7 displays morphings between faces in different positions. As expected, diffAE cannot blend the faces (Line 1). Thus, we consider our neural warping (Line 2) and pass it as input to diffAE, resulting in better interpolations (Line 3).

Our approach also handles faces with varying genders/ethnicities, resulting in high-quality morphings, as shown in Fig 8. It shows that our method learns effective alignments, enabling seamless blendings to preserve details. Morphing in this context is challenging due to feature alignment, and blending skin colors/textures [21]. Additional examples are shown in the supp. material.

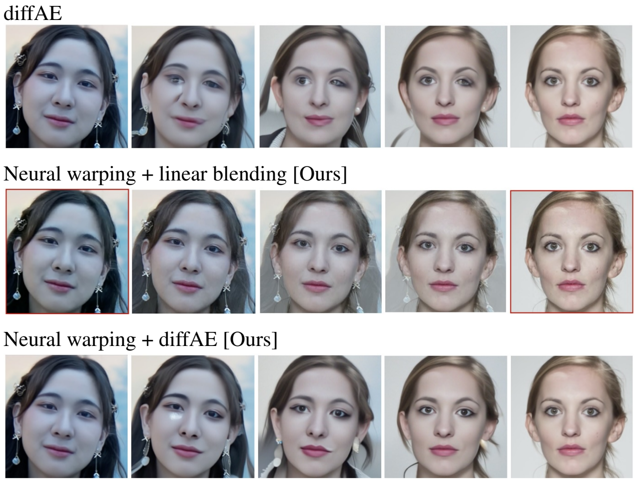

diffAE
Neural warping + linear blending [Ours]
Neural warping + diffAE [Ours]

Figure 7. Morphings between unaligned faces. Columns 1 and 5 are the target images (in red). Columns 2, 3, and 4 are morphings at $t = 0.25, 0.5, 0.75$. Line 1 shows diffAE blending where the target images were cropped to contain mostly the face. Line 2 shows our neural warping and linear blending, and Line 3 shows our neural warping and diffAE blending. Note that the diffAE adds a blurring to the reconstructed images.

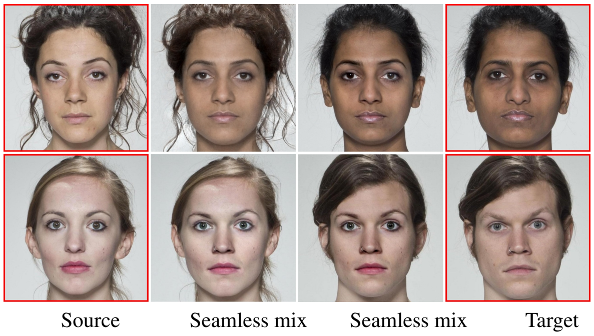


Figure 8. Morphings between faces of different ethnicities (Line 1) and genders (Line 2). Columns 1 and 4 show the target faces. We blend them using seamless mix, at $t = 0.5$, and either the source image as base (Column 2), or the target image as base (Column 3). In both case we employed our neural warping/blending.

Fig 9 shows an example of the warping paths (top) and the linear blending of both images (bottom) created by our method (left) and classic OpenCV warping (right). The creation of a non-linear path lead to a better alignment, and thus a blending with less ghosting artifacts.

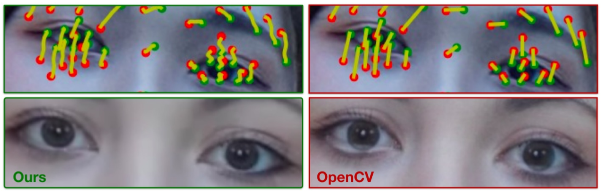


Figure 9. Comparison between our warping (left) and OpenCV (right) and the resulting blendings (bottom row).

Additionally, our method handles morphing between faces with different expressions (Fig 10, top row), partial occlusions (Fig 10, bottom row) and, poses (Fig 11). In Fig 10, we employ linear, our neural Poisson, and diffAE blendings, while in Fig 11 we compare diffAE and MorDiff [6] with our generative blending.

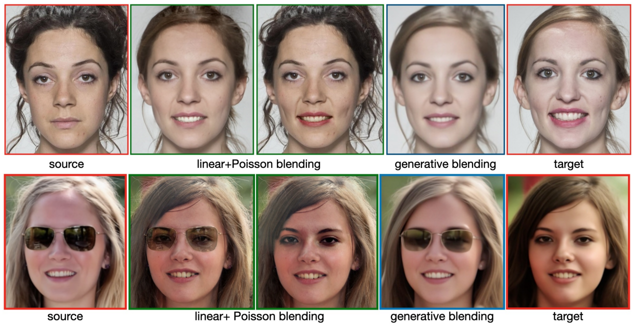


Figure 10. Morphings between subjects with different expressions (top) and, with partial occlusion and faces in the wild (bottom).

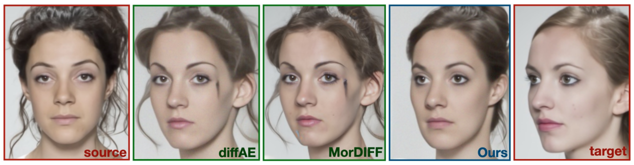


Figure 11. Morphings between faces with different poses.

Feature transfer using neural warping/blending

Our method can be used to transfer features between faces, as shown in Fig 12. To transfer features, we train a warping between two faces, select the region with a desired feature, warp the source face to match the target face, and blend only that region in the gradient domain (Sec 3.4.1).

4.2. Quantitative comparisons

We compare our approach with StyleGAN3, diffAE, and the classic OpenCV procedure. We assess the performance of our neural warping with different blendings: linear, seamless cloning, and mixing. From the 102 images of the FRLL dataset [7], we generated 1220 morphings following the protocol in [30], thus resulting in morphings of similar faces (i.e., same gender, similar ethnicity). Moreover, we used the FFHQ alignment, provided as a stand-alone script by the diffAE[1] to post-process the images (both original and morphed), cropping and resizing them to 256×256 pixels.

To assess the visual fidelity, we used *Fréchet inception distance* (FID) [12] and *learned perceptual image patch similarity* (LPIPS) [35]. FID is employed by generative methods to measure the proximity between the distributions of real and generated images [20]. Lower FID values mean that the distributions are close, thus the generated images

[1] https://github.com/phizaz/diffae/blob/master/align.py

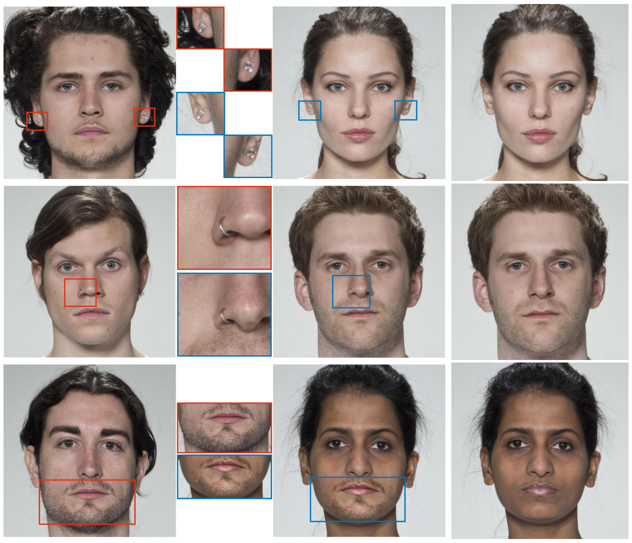


Figure 12. Transference of features between images. Columns 1 and 4 present the source/target faces, Column 2 shows the region containing the desired feature(s) and Column 3 shows the feature(s) transferred to the target image.

are close to the original. LPIPS calculates the similarity of two images by splitting them into patches passed through an image network and measuring their activation similarity. The final LPIPS of the two images is the mean LPIPS of their patches. The FID metric is calculated using `pytorch-fid v0.3.0` [31], while LPIPS uses `lpips v0.1.4` [35].

Table 1 shows the FID and LPIPS scores of the techniques. Here, the target images are I_0 and I_1, and I is the morphing between then at $t = 0.5$. We split the LPIPS score between (I_0, I) and (I, I_1), since the seamless-{clone,mix} blending transfers the warped features of I_1 to I_0, thus leading to a higher similarity between (I_0, I) compared to (I, I_1). Our warping with seamless mix blending achieves higher visual fidelity according to FID and better perceptual similarity to the source image, as indicated by LPIPS (I_0, I), while our method with linear blending obtained LPIPS (I, I_1) comparable to generative methods.

Table 1. FID and LPIPS for OpenCV, StyleGAN3/diffAE, and our warping with different blendings.

Morphing Type	**FID** ↓	**LPIPS** (I_0, I) ↓	**LPIPS** (I, I_1) ↓
OpenCV	68.234	0.275	0.281
StyleGAN3	35.653	0.174	0.173
diffAE	41.356	0.183	0.186
Ours (linear)	31.950	0.158	**0.164**
Ours (S. Clone)	25.290	0.093	0.234
Ours (S. Mix)	**22.604**	**0.081**	0.241
Ours (diffAE)	40,224	0.175	0.176

The results in Table 1 show that by improving the warping, the morphing quality increases (see Lines 1 and 4) such that the resulting images surpass generative methods w.r.t. perceptual metrics. Further improvements in the blending lead to morphings with a natural appearance, and more similar to one of the target images. Additionally, see the morphing-attack-detection (MAD) results in supp. material.

Hardware used The images and morphing networks were trained using an NVIDIA GeForce RTX 3090 GPU, with 24GB of memory. The system has a AMD Ryzen Threadripper PRO 5965WX CPU and 256GB of DDR4 memory.

Ethical Issues One of the problems with face morphing is its use to create fake appearances for official purposes or defamation of individuals. This raises concerns in both the community and the authors. We hope that by exposing our method to the community, we ensure that other colleagues can create detection models to counteract such threats.

Limitations Our method builds a functional representation of the warping to align the features of two faces. It encodes the direct/inverse transformations required in morphing in a single network. Thus, requesting the learning of a non-invertible transformation may lead to inconsistencies. For example, if a particular region of the image collapses during warping, it cannot be inverted. Nevertheless, we can still represent such a transformation with the inverse part of the neural warping or using its direct counterpart.

5. Conclusions

We proposed a face morphing by leveraging coord-based neural networks. We exploited their smoothness to add energy functionals to warp and blend target images seamlessly without the need of derivative discretizations.

Our method ensures continuity in both space and time coordinates, resulting in a smooth transition between images. By operating on a smooth representation of the underlying images, we eliminate the need for pixel interpolation/resampling.The seamless blending of the target images is achieved through the integration of energy functionals, ensuring their harmonious clone. The resulting morphs exhibit a high level of visual fidelity and maintain the overall structure and appearance of the target faces, even when morphing between different genders or ethnicities. Finally, our neural warping offers a versatile framework being easily integrated with generative methods, opening up possibilities for applications in computer graphics and digital entertainment.

In the future, we aim to create morphing datasets using our method to improve MAD models, thus limiting any potential negative impact. We intend to extend it to other type of images, and operate on surfaces as well [19, 22, 23, 32].

Acknowledgements The authors would like to thank projects UIDB/00048/2020[2] and 150991/2023-1 (CNPq and FAPERJ) for partially funding this work. We also thank Daniel Perazzo for helping with the landmark editing UI.

[2] DOI: https://doi.org/10.54499/UIDB/00048/2020

References

[1] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14278–14287, 2021. 2

[2] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. *ACM SIGGRAPH computer graphics*, 26(2):35–42, 1992. 2

[3] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989. 4

[4] Erick Borges, Igor Andrezza, José Marques, Rajiv Mota, and João Primo. Analysis of the eyes on face images for compliance with iso/icao requirements. In *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 173–179, 2016. 3

[5] Robert Carroll, Aseem Agarwala, and Maneesh Agrawala. Image warps for artistic perspective manipulation. In *ACM SIGGRAPH 2010 papers*, pages 1–9. 2010. 1

[6] Naser Damer, Meiling Fang, Patrick Siebke, Jan Niklas Kolf, Marco Huber, and Fadi Boutros. Mordiff: Recognition vulnerability and attack detectability of face morphing attacks created by diffusion autoencoders, 2023. 7

[7] Lisa DeBruine and Benedict Jones. Face research lab london set, 2017. 7

[8] Matteo Ferrara, Annalisa Franco, and Davide Maltoni. The magic passport. In *IJCB 2014 - 2014 IEEE/IAPR International Joint Conference on Biometrics*, 2014. 2

[9] Chris A Glasbey and Kantilal Vardichand Mardia. A review of image-warping methods. *Journal of applied statistics*, 25(2):155–171, 1998. 1, 4

[10] Jonas Gomes, Lucia Darsa, Bruno Costa, and Luiz Velho. *Warping & morphing of graphical objects*. Morgan Kaufmann, 1999. 1

[11] Carla Guerra, Joao Marcos, and Nuno Gonçalves. Automatic validation of icao compliance regarding head coverings: an inclusive approach concerning religious circumstances. In *2023 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, 2023. 3

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 7

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8107–8116. Computer Vision Foundation / IEEE, 2020.

[15] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 852–863. Curran Associates, Inc., 2021. 2

[16] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 6

[17] Alan Lapedes and Robert Farber. Nonlinear signal processing using neural networks: Prediction and system modelling. 1987. 3

[18] Jing Liao, Rodolfo S Lima, Diego Nehab, Hugues Hoppe, Pedro V Sander, and Jinhui Yu. Automating image morphing using structural similarity on a halfway domain. *ACM Transactions on Graphics (TOG)*, 33(5): 1–12, 2014. 2

[19] Hsueh-Ti Derek Liu, Francis Williams, Alec Jacobson, Sanja Fidler, and Or Litany. Learning smooth neural functions via lipschitz regularization. In *ACM SIGGRAPH 2022 Conference Proceedings*. Association for Computing Machinery, 2022. 8

[20] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018. 7

[21] Tom Neubert, Andrey Makrushin, Mario Hildebrandt, Christian Kraetzer, and Jana Dittmann. Extended stirtrace benchmarking of biometric and forensic qualities of morphed face images. *IET Biometrics*, 7(4):325–332, 2018. 6

[22] Tiago Novello, Guilherme Schardong, Luiz Schirmer, Vinícius da Silva, Hélio Lopes, and Luiz Velho. Exploring differential geometry in neural implicits. *Computers & Graphics*, 108:49–60, 2022. 8

[23] Tiago Novello, Vinícius da Silva, Guilherme Schardong, Luiz Schirmer, Hélio Lopes, and Luiz Velho. Neural implicit surface evolution. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14233–14243, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 8

[24] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. Taming the waves: sine as activation function in deep neural networks. 2016. 3

[25] Hallison Paz, Daniel Perazzo, Tiago Novello, Guilherme Schardong, Luiz Schirmer, Vinicius da Silva, Daniel Yukimura, Fabio Chagas, Helio Lopes, and Luiz Velho. Mr-net: Multiresolution sinusoidal neural networks. *Computers & Graphics*, 2023. 2

[26] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pages 313–318. 2003. 5

[27] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 5

[28] K. Raja, M. Ferrara, A. Franco, L. Spreeuwers, I. Batskos, F. Wit, M. Gomez-Barrero, U. Scherhag, D. Fischer, S. Venkatesh, J. M. Singh, G. Li, L. Bergeron, S. Isadskiy, R. Raghavendra, C. Rathgeb, D. Frings, U. Seidel, F. Knopjes, and C. Busch. Morphing Attack Detection - Database, Evaluation Platform and Benchmarking. *IEEE TIFS*, PP:1–1, 2020. 2

[29] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: database and results. *Image and Vision Computing*, 47:3–18, 2016. 300-W, the First Automatic Facial Landmark Detection in-the-Wild Challenge. 6

[30] Eklavya Sarkar, Pavel Korshunov, Laurent Colbois, and Sébastien Marcel. Vulnerability analysis of face morphing attacks from landmarks and generative adversarial networks. *arXiv preprint*, 2020. 7

[31] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, 2020. Version 0.3.0. 8

[32] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3, 6, 8

[33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[34] George Wolberg. Image morphing: a survey. *The visual computer*, 14(8):360–372, 1998. 2

[35] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7, 8